REMARKS

SUMMARY

In the Office Action dated March 15, 2004, the specification was objected to because the section headings were underlined throughout the application. Various claims were objected to due to minor spelling and grammar informalities. Furthermore, claims 1-38 were rejected under 35 U.S.C. §112 and/or 35 U.S.C. §103(a).

The Applicant has amended claims 2, 4, 5, 7, 11, 12, 14, 17, 21, 23, 30, 31, 32, 37, and 38. Claims 1-38 remain currently pending. No new matter has been entered. Reconsideration of the pending claims is respectfully requested in light of these amendments and following remarks.

IN THE SPECIFICATION

The Examiner has objected to the specification under 37 CFR § 1.77(b) because the section headings were underlined throughout the application. The applicant has amended the specification as specified by the Examiner.

IN THE CLAIMS

Claim objections

In the Office Action dated March 15, 2004, claims 2, 4, 5, 7, 8, 11, 12-20, 30, 31, 37, and 38 were objected to because of informalities in the claim language.

Claims 2 and 12 have been amended to correct the spelling of "compressible."

Claims 4, 5, 7, 11, 14, 17, 23, 30, 31, 32, 37, and 38 have been amended to include a colon (":") as the Examiner has requested.

Claims 8, 13, 15, 16, and 18-20 depend from the above amended claims and thus should be in proper form for the objection to be removed.

Claim Rejections Under 35 U.S.C. §112, ¶2

In the Office Action dated March 15, 2004, claims 21-24 were rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the Applicant regards as the invention. In particular, the Examiner stated that the term "data ones" in claims 21 and 23 was vague and indefinite because it was unclear as to what it signified in the claims.

Claims 21 and 23 have been amended to eliminate the term "ones" from "data ones" to render the terms definite as requested by the Examiner. In addition "schema ones" has been replaced by "schemas" to further facilitate clarity of the terms in the claims.

Claims 22 and 24 were also rejected as dependant on claim 21 and 23 respectively. However, with the corrections to the language of claims 21 and 23, these claims now comply with § 112, ¶2.

Accordingly, the Applicant requests that the rejection to claims 21-24 under 35 U.S.C. §112, ¶2 be removed.

Claim Rejections Under 35 U.S.C. §103(a)

Claims 1, 3-5, 9, 11, 13-15, and 19 were rejected under 35 U.S.C. §103(a) as being obvious over US patent No. 6,633,878 issued to Underwood ("Underwood") in view of US Patent No. 6,208,993 issued to Shadmon, et al. ("Shadmon"). Claims 6-8,

16-18, 21, 23, and 32-38 are rejected under 35 U.S.C. § 103(a) as being obvious over Underwood in view of Shadmon, in further view of US Patent No. 6,052,693 issued to Smith et al. ("Smith"). Claims 25-31 are rejected under 35 U.S.C. § 103(a) as being obvious over Underwood in view of Smith. Claims 2 and 12 are rejected under 35 U.S.C. § 103(a) as obvious over Underwood in view of Shadmon, in further view of Smith, and still in further view of US Patent No. 6,604,106 issued to Bodin et al. ("Bodin"). Claims 10, 20, 22, and 24 are rejected under 35 U.S.C. § 103(a) as being obvious over Underwood in view of Shadmon, in further view of Smith, and in further view of US Patent No. 6,651,096 issued to Gai et al. ("Gai").

In response, Applicant respectfully reminds the Examiner that § 103(a) requires the invention being claimed be viewed as a ***whole***. Further, the Courts have held that in an obviousness analysis, the Examiner is to determine the difference between the invention being claimed and the teachings of the closest analogous prior art, and in addition, whether a person of ordinary skill in the art would be motivated to modify these teachings to arrive at the claimed invention.

With respect to claim 1, a novel method for copying/archiving a web-based application is disclosed, which as those of ordinary skill in the art would understand, includes "file system" structures, such as HTML files, scripts, and so forth, and "non-file system" structures, such as data tables, schema of the tables, user lists, and so forth. [The former are referred to as "file system" structures because they are stored as files of a file system, whereas the latter are not.]. More specifically, as set forth by claim 1, the novel method is to include the operations of

> *initializing a file to store said web based application, including creation of a root directory within said file;*

creating data directories under said root directory, and initializing a first plurality of storage data objects under said data directories for all non-file system structures of the web based application; and
copying and storing said non-file system structures into said first plurality of storage data objects.

Thus, claim 1 claims a copying/archiving method for a web-based application, where the web based application is to be copied/archived using ***a file***. Furthermore, the method is to implement its ***own file directory within the file***. Still further, the ***"non-file system" structures*** of the web based application are to be segregated and stored in a set of data objects organized under a particular set of data directories under the root directory of the file directory within the file.

In contrast, Underwood teaches "a system, method, ... for initializing a database used with an *issue tracker*" (Underwood, abs., lns. 1-2), an application subject matter that is separate and distinct from the system administration subject matter of copying/archiving of a web-based application.

Even if we are to ignore the non-analogous subject matters, Underwood teaches the employment of a database, or more specifically, tables of a database, to store the data and application logic of the issue tracker. As those of ordinary skill in the art would appreciate, a file is managed by the file system service of an operating system, which services are limited to the locations of the file, the size of the file, the blocking of the file, and so forth, and the services do not extent to the content of the file. In contrast, a database is managed by a database management subsystem, in which services extend to the content of the databases, including schemas, definitions of the tables and so forth. Accordingly, Underwood's teachings on the employment of a database to store the data and application logic of an issue tracker application do not suggest to a person

of ordinary skill in the art to employ ***a file*** to copy/archive a web-based application having "file system" as well "non-file system" structures.

Given that a database is by definition managed by a database manager, it further follows that the teachings of Underwood do not suggest to a person of ordinary skill to implement a file directory structure within ***the file*** employed to copy/archive the web-based application having "file system" as well "non-file system" structures.

Even if we are to further ignore these deficiencies of Underwood, and assume "tables" may be read as "data objects", Underwood merely teaches the employment of a set of tables for storing data, and another set of tables for application logic; in other words, segregation of data (non-executables) from application logic (executables). It does not teach or suggest the required segregation along the file system line, i.e. between "file system" structures, and "non-file system structures".

In rejecting claim 1, the Examiner specifically cited Fig. 64 and col. 5, lines 59-61 in support of the assertion that Underwood either teaches or suggests the first initialization limitation. Fig. 64 and col. 5 lines 59-61, merely illustrate and describe a conventional directory of a file system. It does not teach or suggest self implementation of a directory system ***within a file***.

The Examiner relied on the abstract and Fig 1.2, col 19, lines 39-44 in support of the assertion that Underwood either teaches or suggests the second initialization limitation. Neither the assertion of "table" being readable as "non-file system structures", nor the description of "first and second recordsets being mappable to business objects" teach or suggest the essence of the second initialization limitation,

which requires the initialization of a separate set of data objects for the segregated storage of the "non-file system" structures of the web based applications.

Thus, for at least the above reasons, and the acknowledged deficiency of Underwood, claim 1 is non-obvious and patentable over Underwood. Shadmon, et al does not cure the above discussed deficiencies of Underwood, therefore claim 1 remains patentable over Underwood, even when combined with Shadmon.

Claims 3-5 and 9 depend from claim 1, and as a result of such dependency are also patentable over Underwood in view of Shadmon.

Claim 11 also stands rejected under § 103(a) over Underwood in view of Shadmon. Because claim 11 has similar novel elements to claim 1, the above remarks regarding claim 1 apply to claim 11 as well. Thus, for the at least the above reasons, and the acknowledged deficiency of Underwood, claim 11 is non-obvious and patentable over Underwood. Shadmon, et al does not cure the above discussed deficiencies of Underwood, therefore claim 11 remains patentable over Underwood, even when combined with Shadmon. As such, the applicant submits that claim 11 is in proper form for allowance and requests the § 103(a) rejection be removed.

Claims 13-15 and 19 depend from claim 11, and as a result of such dependency are also patentable over Underwood in view of Shadmon.

Claims 6-8 and 16-18 were rejected over Underwood in view of Shadmon in further view of Smith. Smith is cited for teaching various aspects related to claims 6-8 and 16-18. Claims 6-8 and 16-18 depend from claims 1 and 11 respectively. Smith does not cure the deficiency of Underwood or Shadmon. Smith does not disclose or suggest, among other things, *initializing a file to store said web based application,*

including creation of a root directory within said file. Thus, for at least the reasons discussed above with respect to claims 1 and 11, Applicant respectfully submits that claims 6-8 and 16-18 are patentable over Underwood in view of Shadmon in further view of Smith.

Claims 2 and 12 were rejected over Underwood in view of Shadmon in further view of Smith in further view of Bodin. As previously stated Smith does not cure the deficiency of Underwood and Shadmon. Bodin is cited for teaching compression. However, Bodin does not disclose or suggest, among other things, *initializing a file to store said web based application, including creation of a root directory within said file*. Thus, for at least the reasons discussed above with respect to claims 1 and 11, Applicant respectfully submits that claims 2 and 12 are patentable over Underwood in view of Shadmon in further view of Smith in further view of Bodin.

Claims 10 and 20 were rejected over Underwood in view of Shadmon in further view of Smith in further view of Gai. As previously stated Smith does not cure the deficiency of Underwood and Shadmon. Gai is cited for teaching access of control lists. However, Gai does not disclose or suggest, among other things, *initializing a file to store said web based application, including creation of a root directory within said file*. Thus, for at least the reasons discussed above with respect to claims 1 and 11, Applicant respectfully submits that claims 10 and 20 are patentable over Underwood in view of Shadmon in further view of Smith in further view of Gai.

With respect to method claim 21 and apparatus claim 23, which stand rejected under § 103(a) over Underwood in view of Shadmon in further view of Smith, a novel method of copying/restoring a web based application into a domain, and an apparatus

directed at carrying out the novel method are disclosed. As set forth in claim 21, and similarly set forth in claim 23, the novel method is to include:

> retrieving a structural description describing non-file system structures and files of the web based application;
> determining in accordance with at least said structural description non-file system structures of the web based application, including constitutions of the non-file system structures, and files of the web based application, including pathnames of the files;
> retrieving schemas and data of said non-file system structures in accordance with the result of said determination;
> storing said data of said non-file system structures in accordance with schemas of said non-file system structures; and
> retrieving and storing said files in accordance with the result of said determination.

Thus, claims 21 and 23 claim a copying/restoring method for a web-based application, where the web-based application is to be copied/restored using a ***structural description***. Additionally, this structural description describes both ***"non-file system" structures*** and ***files*** of the web based application. Further, the structural description is used to determine constitutions of the non-file system structures, and files of the web based application and retrieve schemas and data of the non-file system structures. Still further, the data is stored in accordance with the schemas, and the segregated files are retrieved and stored in accordance with the above determination.

The Examiner points to the following passages in Underwood as teaching *retrieving a structural description describing non-file system structures and files of a web based application*:

> Encapsulation enforces data abstraction through the organization of data into small, independent objects that can communicate with each other. Encapsulation protects the data in an object from accidental damage, but allows other objects to interact with that data by calling the object's member functions and structures.

Underwood, col. 13, lines. 43-49.

> Application frameworks reduce the total amount of code that a programmer has to write from scratch. However, because the framework is really a generic application that displays windows, supports copy and paste, and so on, the programmer can also relinquish control to a greater degree than event loop programs permit. The framework code takes care of almost all event handling and flow of control, and the programmer's code is called only when the framework needs it (e.g. to create or manipulate a proprietary data structure).

Underwood, col. 15, lines. 5-14.

Instead of teaching a method of retrieving a structural description of non-file system structures and files, these passages merely disclose a method of protecting data by encapsulating it and allowing the encapsulating objects to communicate by calling the object's member functions and structures. Additionally, the second passage discusses application framework code that takes care of event handling and flow of control. Data encapsulation and framework code both involve the manipulation of data, but neither one discloses or suggests retrieving a structural description describing non-file system structures and files of the web based application.

Shadmon is cited for teaching one or more elements of claim 21. Assuming, arguendo, that Shadmon teaches these element(s), Shadmon does not cure the deficiency of Underwood. That is Shadmon does not disclose or suggest retrieving a structural description describing non-file system structures and files of the web based application. Similarly, Smith is citing for teaching one or more elements of claim 21. Assuming, arguendo, that Smith teaches these element(s), Smith does not cure the deficiency of Underwood. That is Smith does not disclose or suggest retrieving a structural description describing non-file system structures and files of the web based application.

Thus, for at least the above reasons, and the acknowledged deficiency of Underwood, claims 21 and 23 are non-obvious and patentable over Underwood in view of Shadmon in further view of Smith.

Claims 22 and 24 were rejected over Underwood in view of Shadmon in further view of Smith in further view of Gai. As previously stated Smith does not cure the deficiency of Underwood. Gai is cited for teaching access of control lists. However, Gai does not disclose or suggest, among other things, retrieving a structural description describing non-file system structures and files of the web based application. Thus, for at least the reasons discussed above with respect to claims 21 and 23, Applicant respectfully submits that claims 22 and 24 are patentable over Underwood in view of Shadmon in further view of Smith in further view of Gai.

Claims 25 stands rejected under § 103 over Underwood in view of Smith. Claim 25 is also directed towards a method of copying/restoring a web based application into a domain. More specifically, this method comprises:

> retrieving a plurality of data table schemas for a plurality of data tables of the web based application, and data of the data tables;
> as each data table schema is retrieved,
> storing the data table schema in a temporal storage location,
> creating a data table in accordance with the data table schema,
> determining if data for the data table has already been retrieved,
> storing the data into the data table if the data for the data table has already been retrieved; and
> as each collection of data for a data table is retrieved,
> storing the collection of data in a temporal storage location,
> determining if the data table has already been created,
> storing the data into the data table if the data table has already been created.

Thus, claim 25 of the present invention discloses ***retrieving*** a plurality of data table schemas and data for a plurality of data tables. Further, as the schemas are retrieved, the method discloses storing the schemas and data in temporal storage locations, ***creating a data table in accordance with the schemas***, determining if the data has already been retrieved, and if not, storing the data in the data tables. Additionally, as each collection of data is retrieved, the method specifies storing the data in a temporal location, determining if the data table has been created, and if so copying the data into the data table.

Underwood is cited for teaching a method for copying/restoring a web based application into a domain (column 14, lines 23-27 and column 55, lines 61-67). However, Underwood does not teach a method for copying/restoring a web based application into a domain. As cited Underwood discloses:

> Class libraries are very flexible. As programs grow more complex, more programmers are forced to reinvent basic solutions to basic problems over and over again. A relatively new extension of the class library concept is to have a framework of class libraries. This framework is more complex and consists of significant collections of collaborating classes that capture both the small scale patterns and major mechanisms that implement the common requirements and design in a specific application domain. They were first developed to free application programmers from the chores involved in displaying menus, windows, dialog boxes, and other standard user interface elements for personal computers.

Underwood, col. 14, lines 23-27.

> The AFPLExtent class provides the mapping between the business object and its associated database table. In addition, the AFPLExtent class represents the domain defined by the visible part of the database table for the specified user. This class holds the passed in database URL, username and password used during the access to the database. Lastly, the AFPLExtent class manages the database connection.

Underwood, col. 55, lines 61-67.

Thus, Underwood simply does not disclose retrieving a plurality of retrieving a plurality of data table schemas for a plurality of data tables of the web based application, and data of the data tables. Smith is cited for teaching retrieval of schemas. Even assuming that Smith does so teach, Smith does not cure the deficiency of Underwood. That is, Smith does not disclose or suggest method for copying/restoring a web based application into a domain. Thus, Applicant respectfully submits that claim 25 is patentable over Underwood in view of Smith.

Claims 26 - 31 depend from claim 25 and by nature of there dependency from claim 25, for at least the reasons discussed above with claim 25, claims 26-31 are patentable over Underwood in view of Smith.

Claim 32 stands rejected under § 103 as obvious over Underwood in view of Shadmon, in further view of Smith. Like claim 25, claim 32 contains as an element retrieving a plurality of data table schemas for a plurality of data tables of a web based application, and data of the data tables. The Examiner again stated that this element is taught by Underwood in view of Smith. However, following at least the reasoning set out above for claim 25, the applicant submits that both Underwood and Smith fails to teach this element. Further the cited additional reference Shadmon does not disclose or suggest retrieving a plurality of data table schemas for a plurality of data tables of a web based application, and data of the data tables. Thus, Applicant respectfully submits that claim 32 is patentable over Underwood in view of Smith in further view of Shadmon. Claims 33-38 depend from claim 32 and by nature of there dependency from claim 32, for at least the reasons discussed above with claim 32, claims 33-38 are patentable over Underwood in view of Smith in further view of Shadmon.

CONCLUSION

In light of the above amendments and remarks, this application is now in condition for allowance. Claims 1-38 remain currently pending. Early issuance of Notice of Allowance is respectfully requested.

The Commissioner is hereby authorized to charge shortages or credit overpayments to Deposit Account No. 500393. A Fee Transmittal is enclosed in duplicate for fee processing purposes.

Respectfully submitted,
SCHWABE, WILLIAMSON & WYATT, P.C.

Dated: 15 Jun 04

Robert Watt
Registration No. 45,890

Pacwest Center, Suites 1600-1900
1211 SW Fifth Avenue
Portland, Oregon 97204
Telephone: 503-222-9981